



Full credit is given to the above companies including the OS that this PDF file was generated!

Red Hat Enterprise Linux Release 9.2 Manual Pages on 'ftok.3p' command

\$ man ftok.3p

FTOK(3P) POSIX Programmer's Manual FTOK(3P)

PROLOG

This manual page is part of the POSIX Programmer's Manual. The Linux implementation of this interface may differ (consult the corresponding Linux manual page for details of Linux behavior), or the interface may not be implemented on Linux.

NAME

ftok ? generate an IPC key

SYNOPSIS

```
#include <sys/ipc.h>

key_t ftok(const char *path, int id);
```

DESCRIPTION

The ftok() function shall return a key based on path and id that is usable in subsequent calls to msgget(), semget(), and shmget(). The application shall ensure that the path argument is the pathname of an existing file that the process is able to stat(), with the exception that if stat() would fail with [E_OVERFLOW] due to file size, ftok() shall still succeed.

The ftok() function shall return the same key value for all paths that name the same file, when called with the same id value, and should return different key values when called with different id values or with paths that name different files existing on the same file system at the same time. It is unspecified whether ftok() shall return the same key

value when called again after the file named by `path` is removed and recreated with the same name.

Only the low-order 8-bits of `id` are significant. The behavior of `ftok()` is unspecified if these bits are 0.

RETURN VALUE

Upon successful completion, `ftok()` shall return a key. Otherwise, `ftok()` shall return `(key_t)-1` and set `errno` to indicate the error.

ERRORS

The `ftok()` function shall fail if:

EACCES Search permission is denied for a component of the path prefix.

EIO An error occurred while reading from the file system.

ELOOP A loop exists in symbolic links encountered during resolution of the path argument.

ENAMETOOLONG

The length of a component of a pathname is longer than `{NAME_MAX}`.

ENOENT A component of `path` does not name an existing file or path is an empty string.

ENOTDIR

A component of the path prefix names an existing file that is neither a directory nor a symbolic link to a directory, or the path argument contains at least one non-`<slash>` character and ends with one or more trailing `<slash>` characters and the last pathname component names an existing file that is neither a directory nor a symbolic link to a directory.

The `ftok()` function may fail if:

ELOOP More than `{SYMLOOP_MAX}` symbolic links were encountered during resolution of the path argument.

ENAMETOOLONG

The length of a pathname exceeds `{PATH_MAX}`, or resolution of a symbolic link produced an intermediate result with a length that exceeds `{PATH_MAX}`.

The following sections are informative.

EXAMPLES

Getting an IPC Key

The following example gets a key based on the pathname /tmp and the ID value a. It also assigns the value of the resulting key to the semkey variable so that it will be available to a later call to semget(), msgget(), or shmget().

```
#include <sys/ipc.h>
...
key_t semkey;
if ((semkey = ftok("/tmp", 'a')) == (key_t) -1) {
    perror("IPC error: ftok"); exit(1);
}
```

APPLICATION USAGE

For maximum portability, id should be a single-byte character.

Applications should not assume that the resulting key value is unique.

RATIONALE

None.

FUTURE DIRECTIONS

Future versions of this standard may add new interfaces to provide unique keys.

SEE ALSO

msgget(), semget(), shmget()

The Base Definitions volume of POSIX.1-2017, <sys_ipc.h>

COPYRIGHT

Portions of this text are reprinted and reproduced in electronic form from IEEE Std 1003.1-2017, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 7, 2018 Edition, Copyright (C) 2018 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between this version and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document. The original Standard can be obtained online at <http://www.opengroup.org/unix/online.html>.

Any typographical or formatting errors that appear in this page are most likely to have been introduced during the conversion of the source files to man page format. To report such errors, see https://www.ker?nel.org/doc/man-pages/reporting_bugs.html .

IEEE/The Open Group

2017

FTOK(3P)